\ Shunting-yard algorithm in Forth

\ convert infix expression to reverse polish notation

\ Output is an RPN text string that can be interpreted or compiled

\ while there are tokens to be read:

\ read a token.

CASE

\ if the token is a number, then push it to the output queue.

\ if the token is an operator, then:

\ while there is an operator at the top of the operator stack with

\ greater than or equal to precedence and the operator is left associative:

\ pop operators from the operator stack, onto the output queue.

\ push the read operator onto the operator stack.

\ if the token is a left bracket (i.e. "("), then:

\ push it onto the operator stack.

\ if the token is a right bracket (i.e. ")"), then:

\ while the operator at the top of the operator stack is not a left bracket:

\ pop operators from the operator stack onto the output queue.

\ pop the left bracket from the stack.

\ \*\*\* IF THE STACK RUNS OUT WITHOUT FINDING A LEFT BRACKET, THEN THERE ARE MISMATCHED PARENTHESES. \*\*\*

\ if there are no more tokens to read:

\ while there are still operator tokens on the stack:

\ \*\*\* IF THE OPERATOR TOKEN ON THE TOP OF THE STACK IS A BRACKET, THEN THERE ARE MISMATCHED PARENTHESES. \*\*\*

\ pop the operator onto the output queue.

\ exit.

: INFIX ( addr len -- )

OUTQ EMPTY

COLLAPSE

BEGIN

READ\_TOKEN C@ 0>

WHILE

CASE

OPERATOR? ?OF

CR ." OP! " .TOKEN

BEGIN

OPSTACK?

WHILE

TOP$ >OUTQ DROP$

REPEAT

TOKEN$ SPUSH ENDOF

ISNUMBER? ?OF CR ." Num! " .TOKEN

TOKEN$ >OUTQ ENDOF

TOKEN$ '(' MATCH IF CR '(' TYPE .TOKEN

TOKEN$ $PUSH ENDOF

TOKEN$ ')' MATCH IF CR ')' TYPE .TOKEN

POP-UNTIL-LEFT ENDOF

( DEFAULT) -1 ABORT" INFIX: bad expression"

ENDCASE

REPEAT

OPSTACK->OUTQ

COLLAPSE ;